Package ‘SurrogateParadoxTest’

June 13, 2025

Type Package

Title Empirical Testing of Surrogate Paradox Assumptions

Version 2.1

Date 2025-06-11

Description Provides functions to nonparametrically assess assumptions necessary to prevent the surrogate paradox through hypothesis tests of stochastic dominance, monotonicity of regression functions, and non-negative residual treatment effects. Details are described in: Hsiao E, Tian L, and Parast L (2025). ``Avoiding the surrogate paradox: an empirical framework for assessing assumptions.'' Journal of Nonparametric Statistics <doi:10.1080/10485252.2025.2498609>. Additionally, there are functions to assess resilience to the surrogate paradox via calculation of the resilience probability, the resilience bound, and the resilience set. Details will be available in Hsiao E, Tian L, and Parast L, ``Resilience Measures for the Surrogate Paradox'' (Under Review). A tutorial for this package can be found at <`https://www.laylaparast.com/surrogateparadoxtest`>.

License GPL

Imports stats, MonotonicityTest, MASS, ggplot2

NeedsCompilation no

Author Emily Hsiao [aut, cre]

Maintainer Emily Hsiao <`ehsiao@utexas.edu`>

Contents

`a_b_hat` *Helper function*

Description

Helper function for monotonicity test

Usage

```
a_b_hat(r, s, X, Y)
```

Arguments

`r`	Index to start summation.
`s`	Index to end summation.
`X`	Vector of X values.
`Y`	Vector of Y values.

Value

`a_hat`	Numeric value of a_hat
`b_hat`	Numeric value of b_hat

Author(s)

Emily Hsiao

`barrett_donald_cutoff` *Helper function for stochastic dominance test*

Description

Rejection cutoff value for stochastic dominance value based on alpha level.

Usage

```
barrett_donald_cutoff(alpha)
```

Arguments

`alpha`	Desired alpha level for stochastic dominance test.

Value

Cutoff value for stochastic dominance test

Author(s)

Emily Hsiao

`barrett_donald_p` *Helper function for stochastic dominance test*

Description

Calculates p-value of the stochastic dominance test

Usage

```
barrett_donald_p(statistic)
```

Arguments

`statistic`	Test statistic calculated in stochastic dominance test

Value

p-value of the test statistic

Author(s)

Emily Hsiao

`calculate_bandwidth` *Helper function for kernel smoother*

Description

Calculates the appropriate bandwidth for Nadaraya-Watson kernel smoother.

Usage

```
calculate_bandwidth(s)
```

Arguments

`s`	Vector of surrogate values.

Value

Desired bandwidth for kernel smoother.

Author(s)

Emily Hsiao

`fourier_interval` *Fourier Resilience Interval*

Description

Calculates the resilience probability and resilience bound, assuming that the mean function of Study B data is generated by adding a Fourier basis with random coefficients with specified parameters.

Usage

```
fourier_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, var_vec, period,
n.iter = 500, M = 100, q_quant = 0.1,      plot = FALSE, intervals
= TRUE, get_var = TRUE)
```

Arguments

`s0.A`	Vector of surrogate values in the control group of Study A.
`y0.A`	Vector of primary outcome values in the control group of Study A.
`s1.A`	Vector of surrogate values in the treatment group of Study A.
`y1.A`	Vector of primary outcome values in the treatment group of Study A.
`s0.B`	Vector of surrogate values in the control group of Study B.
`s1.B`	Vector of surrogate values in the treatment group of Study B.
`var_vec`	Length-4 vector governing the variance of the random coefficients of the mean function for Study B.
`period`	Length-3 vector dictating the period of th Fourier basis for Study B. Each item represents a proportion of the range of the surrogate values of Study A.
`n.iter`	Number of Δ_B samples to generate.
`M`	Number of bootstrap iterations to estimate the SE of $\hat{p}_0$ and $\hat{q}_\alpha$.
`q_quant`	Desired quantile for the resilience bound. Default is 0.10.
`plot`	TRUE or FALSE; include plots of randomly generated functions in results. Default is FALSE.
`intervals`	TRUE or FALSE; return the inner 95% of Δ_B samples. Default is TRUE.
`get_var`	TRUE or FALSE; return an estimated variance of the $\hat{p}_0$ and $\hat{q}_\alpha$ parameters. Default is TRUE.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

`Delta_hats`	Vector of samples of Δ_B.
`Delta_estimate`	Mean of samples of Δ_B.
`p_hat`	Estimated value $\hat{p}_0$, the probability of the surrogate paradox.
`q_hat`	Estimated value $\hat{q}_\alpha$

, the resilience bound.

`p_se` Estimated variance of $\hat{p}_0$; only returned if get_var = TRUE.

`q_se` Estimated variance of $\hat{q}_\alpha$; only returned if get_var = TRUE.

`control_plot` Plot including data points of control group of Study A and functions generated for Study B; only returned if plot = TRUE.

`treatment_plot` Plot including data points of treatment group of Study A and functions generated for Study B; only returned if plot = TRUE.

Author(s)

Emily Hsiao

References

Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Examples

```
s0.A <- rnorm(20, 3, sqrt(3))
y0.A <- sapply(s0.A, function(x) 2 * x - 1)
s1.A <- rnorm(20, 4, sqrt(3))
y1.A <- sapply(s1.A, function(x) x + 3)
s0.B <- rnorm(20, 4.75, sqrt(1))
s1.B <- rnorm(20, 5.25, sqrt(1))
result <- fourier_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B,
var_vec = c(0.25, 0.25, 0.1, 0.1), period = c(0.5, 0.25, 0.1), n.iter = 20, M = 20)
result$p_hat
```

fourier_resilience_set *Fourier Resilience Set*

Description

Creates a plot of the resilience set i.e., the possible variance parameters of the coefficients of the Fourier terms of the mean function of Study B such that the probability of the surrogate paradox is below a threshold α. Note that this function assumes that the covariance matrix of the random coefficients takes the form $\Sigma = diag(\sigma_{11}^2, \sigma_{11}^2, \sigma_{22}^2, \sigma_{22}^2)$.

Usage

```
fourier_resilience_set(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B,
sig1_values, sig2_values, alpha = 0.10)
```

Arguments

`s0.A` Vector of surrogate values in the control group of Study A.

`y0.A` Vector of primary outcome values in the control group of Study A.

`s1.A` Vector of surrogate values in the treatment group of Study A.

`y1.A` Vector of primary outcome values in the treatment group of Study A.

`s0.B` Vector of surrogate values in the control group of Study B.

`s1.B` Vector of surrogate values in the treatment group of Study B.

`sig1_values` Vector of values of σ_{11}^2 for which the probability of surrogate paradox should be calculated with the given data.

`sig2_values` Vector of values of σ_{22}^2 for which the probability of surrogate paradox should be calculated with the given data.

`alpha` Threshold value of the surrogate paradox. Default is 0.10.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

Returns a ggplot object with values of σ_{11}^2 on the x-axis and σ_{22}^2 on the y-axis, and regions where $P(\Delta_B) < \alpha$ with given data and parameters highlighted in blue.

Author(s)

Emily Hsiao

References

Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Examples

```
n_A=200
n_B=200
s0.A <- rnorm(n_A, mean = 10, sd = 2)
y0.A <- 5 + 0.5 * s0.A + rnorm(n_A, mean = 0, sd = 0.5)

s1.A <- rnorm(n_A, mean = 12, sd = 2.5)
y1.A <- 5.5 + 0.6 * s1.A + rnorm(n_A, mean = 0, sd = 0.6)

 s0.B <- rnorm(n_B, 10, 2)
 s1.B <- rnorm(n_B, 11, 3)
 sig1_vals_example <- seq(0.1, 1, length.out = 30) # Sigma squared values
 sig2_vals_example <- seq(0.1, .5, length.out = 30)

 # Run the function with the generated example data

 fourier_resilience_set(
   s0.A, y0.A, s1.A, y1.A, s0.B, s1.B,
   sig1_values = sig1_vals_example,
   sig2_values = sig2_vals_example,
   alpha = 0.05
 )
```

`gaussian_kernel` *Helper function for kernel smoother*

Description

Gaussian kernel used for kernel smoother.

Usage

```
gaussian_kernel(x)
```

Arguments

`x`	x

Value

Gaussian kernel applied to x

Author(s)

Emily Hsiao

`gaussian_process_interval` *Gaussian Process Resilience Interval*

Description

Calculates the resilience probability and resilience bound, assuming that the mean function of Study B data is generated according to a Gaussian Process with specified parameters.

Usage

```
gaussian_process_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sigma2, theta,
  n.iter = 500, M = 100, q_quant = 0.1, plot = FALSE, intervals = TRUE, get_var = TRUE)
```

Arguments

`s0.A`	Vector of surrogate values in the control group of Study A.
`y0.A`	Vector of primary outcome values in the control group of Study A.
`s1.A`	Vector of surrogate values in the treatment group of Study A.
`y1.A`	Vector of primary outcome values in the treatment group of Study A.
`s0.B`	Vector of surrogate values in the control group of Study B.
`s1.B`	Vector of surrogate values in the treatment group of Study B.
`sigma2`	Variance parameter of the Gaussian Process.
`theta`	Smoothness parameter of the Gaussian Process.
`n.iter`	Number of Δ_B samples to generate.

`M` Number of bootstrap iterations to estimate the SE of $\hat{p}_0$ and $\hat{q}_\alpha$.

`q_quant` Desired quantile for the resilience bound. Default is 0.10.

`plot` TRUE or FALSE; include plots of randomly generated functions in results. Default is FALSE.

`intervals` TRUE or FALSE; return the inner 95% of Δ_B samples. Default is TRUE.

`get_var` TRUE or FALSE; return an estimated variance of the $\hat{p}_0$ and $\hat{q}_\alpha$ parameters. Default is TRUE.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

`Delta_hats` Vector of samples of Δ_B.

`Delta_estimate` Mean of samples of Δ_B.

`p_hat` Estimated value $\hat{p}_0$, the probability of the surrogate paradox.

`q_hat` Estimated value $\hat{q}_\alpha$

, the resilience bound.

`p_se` Estimated variance of $\hat{p}_0$; only returned if get_var = TRUE.

`q_se` Estimated variance of $\hat{q}_\alpha$; only returned if get_var = TRUE.

`control_plot` Plot including data points of control group of Study A and functions generated for Study B; only returned if plot = TRUE.

`treatment_plot` Plot including data points of treatment group of Study A and functions generated for Study B; only returned if plot = TRUE.

`p_interval` Inner 95% of $\hat{p}_0$ samples from bootstrap; only returned if intervals = TRUE.

`q_interval` Inner 95% of $\hat{q}_\alpha$ samples from bootstrap; only returned if intervals = TRUE.

Author(s)

Emily Hsiao

References

Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Examples

```
s0.A <- rnorm(20, 3, sqrt(3))
y0.A <- sapply(s0.A, function(x) 2 * x - 1)
s1.A <- rnorm(20, 4, sqrt(3))
y1.A <- sapply(s1.A, function(x) x + 3)
s0.B <- rnorm(20, 4.75, sqrt(1))
s1.B <- rnorm(20, 5.25, sqrt(1))
result <- gaussian_process_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sigma2 = 0.25,
  theta = 2, n.iter = 20, M = 20)
result$p_hat
```

gp_resilience_set *Gaussian Process Resilience Set*

Description

Creates a plot of the resilience set i.e., the possible parameters of the Gaussian Process such that the probability of the surrogate paradox is below a threshold α.

Usage

```
gp_resilience_set(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sigma2_vals, theta_vals,
  alpha = 0.10)
```

Arguments

s0.A	Vector of surrogate values in the control group of Study A.
y0.A	Vector of primary outcome values in the control group of Study A.
s1.A	Vector of surrogate values in the treatment group of Study A.
y1.A	Vector of primary outcome values in the treatment group of Study A.
s0.B	Vector of surrogate values in the control group of Study B.
s1.B	Vector of surrogate values in the treatment group of Study B.
sigma2_vals	Vector of values of σ^2 for which the probability of surrogate paradox should be calculated with the given data.
theta_vals	Vector of values of θ for which the probability of surrogate paradox should be calculated with the given data.
alpha	Threshold value of the surrogate paradox. Default is 0.10.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

Returns a ggplot object with values of θ on the x-axis and σ^2 on the y-axis, and regions where $P(\Delta_B) < \alpha$ with given data and parameters highlighted in blue.

Author(s)

Emily Hsiao

References

Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Examples

```
n_A=200
n_B=200
s0.A <- rnorm(n_A, mean = 10, sd = 2)
y0.A <- 5 + 0.5 * s0.A + rnorm(n_A, mean = 0, sd = 0.5)

s1.A <- rnorm(n_A, mean = 12, sd = 2.5)
y1.A <- 5.5 + 0.6 * s1.A + rnorm(n_A, mean = 0, sd = 0.6)

 s0.B <- rnorm(n_B, 10, 2)
 s1.B <- rnorm(n_B, 11, 3)


  sigma2_values <- seq(0.01, 10, length.out = 30)
  theta_values <- seq(0.01, 10, length.out = 30)

  gp_resilience_set(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sigma2_values, theta_values,
  alpha = 0.05)
```

modified_S_stat *Helper function for non-negative residual treatment effect test.*

Description

Calculates the value of the test statistic in the NNR test.

Usage

```
modified_S_stat(mu0_hat, mu1_hat, s0, y0, s1, y1, grid_x, boot = FALSE)
```

Arguments

`mu0_hat` Kernel-smoothed estimate of μ_0 function.

`mu1_hat` Kernel-smoothed estimate of μ_1 function.

`s0` Vector of surrogate values in control group.

`y0` Vector of endpoint values in control group.

`s1` Vector of surrogate values in treatment group.

`y1` Vector of endpoing values in the treatment group.

`grid_x` Values of s over which supremum is calculated.

`boot` Whether this is a bootstrapped statistic or the test statistic.

Value

`s_hat` Value of the test statistic

`sup` Value of the supremum over grid_x

Author(s)

Emily Hsiao

monotonicity_test *Monotonicity test*

Description

Runs the test of monotonicity for a regression function.

Usage

```
monotonicity_test(X, Y, h = NA, m = 5, bootstrap_n = 100,
alpha = 0.05)
```

Arguments

X	Vector of X values.
Y	Vector of Y values.
h	Bandwidth for the kernel smoother.
m	Window size to calculate linear regression.
bootstrap_n	Desired number of bootstrap samples.
alpha	Desired alpha level of the test.

Value

T_m_value	Value of the test statistic.
p_val	p-value of test
reject	whether the test rejects the null
T_m_samples	Vector of bootstrapped values of test statistic

Author(s)

Emily Hsiao

References

Hall, Peter, and Nancy E. Heckman. "Testing for monotonicity of a regression mean by calibrating for linear functions." Annals of Statistics (2000): 20-39.

nnr_test *Non-negative residual treatment effect function*

Description

Runs the test of non-negative residual treatment effect given two sets of surrogate and primary endpoint values.

Usage

```
nnr_test(s0, y0, s1, y1, n_bootstrap = 200, alpha = 0.05)
```

Arguments

s0	Vector of surrogate values in control group.
y0	Vector of endpoint values in control group.
s1	Vector of surrogate values in treatment group.
y1	Vector of endpoint values in treatment group.
n_bootstrap	Desired number of bootstrap samples.
alpha	Desired alpha level of test.

Value

p_value	p-value of test.
reject	Whether the test rejects the null hypothesis.
s_hat	Calculated value of test statistic.
s_vec	Vector of bootstrapped values of test statistic.

Author(s)

Emily Hsiao

References

Hsiao et all 2024 (Under review)

polynomial_interval *Polynomial Resilience Interval*

Description

Calculates the resilience probability and resilience bound, assuming that the mean function of Study B data is generated by adding a polynomial basis with random coefficients with specified parameters.

Usage

```
polynomial_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, var_vec, n.iter = 500,
M = 100, q_quant = 0.1, plot =    FALSE, intervals = TRUE, get_var = TRUE)
```

Arguments

`s0.A` Vector of surrogate values in the control group of Study A.

`y0.A` Vector of primary outcome values in the control group of Study A.

`s1.A` Vector of surrogate values in the treatment group of Study A.

`y1.A` Vector of primary outcome values in the treatment group of Study A.

`s0.B` Vector of surrogate values in the control group of Study B.

`s1.B` Vector of surrogate values in the treatment group of Study B.

`var_vec` Length-4 vector governing the variance of the random coefficients of the mean function for Study B.

`n.iter` Number of Δ_B samples to generate.

`M` Number of bootstrap iterations to estimate the SE of $\hat{p}_0$ and $\hat{q}_\alpha$.

`q_quant` Desired quantile for the resilience bound. Default is 0.10.

`plot` TRUE or FALSE; include plots of randomly generated functions in results. Default is FALSE.

`intervals` TRUE or FALSE; return the inner 95% of Δ_B samples. Default is TRUE.

`get_var` TRUE or FALSE; return an estimated variance of the $\hat{p}_0$ and $\hat{q}_\alpha$ parameters. Defaults to TRUE.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

`Delta_hats` Vector of samples of Δ_B.

`Delta_estimate` Mean of samples of Δ_B.

`p_hat` Estimated value $\hat{p}_0$, the probability of the surrogate paradox.

`q_hat` Estimated value $\hat{q}_\alpha$

, the resilience bound.

`p_se` Estimated variance of $\hat{p}_0$; only returned if get_var = TRUE.

`q_se` Estimated variance of $\hat{q}_\alpha$; only returned if get_var = TRUE.

`control_plot` Plot including data points of control group of Study A and functions generated for Study B; only returned if plot = TRUE.

`treatment_plot` Plot including data points of treatment group of Study A and functions generated for Study B; only returned if plot = TRUE.

Author(s)

Emily Hsiao

References

Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Examples

```
s0.A <- rnorm(20, 3, sqrt(3))
y0.A <- sapply(s0.A, function(x) 2 * x - 1)
s1.A <- rnorm(20, 4, sqrt(3))
y1.A <- sapply(s1.A, function(x) x + 3)
s0.B <- rnorm(20, 4.75, sqrt(1))
s1.B <- rnorm(20, 5.25, sqrt(1))
result <- polynomial_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B,
var_vec = c(0.25, 0.25, 0.1, 0.1),n.iter = 20, M = 20)
result$p_hat
```

---

polynomial_resilience_set

Polynomial Resilience Set

Description

Creates a plot of the resilience set i.e., the possible variance parameters of the coefficients of the polynomial terms of the mean function of Study B such that the probability of the surrogate paradox is below a threshold α. Note that this function assumes that the covariance matrix of the random coefficients takes the form $\Sigma = diag(\sigma^2_{11}, \sigma^2_{11}, \sigma^2_{22}, \sigma^2_{22})$.

Usage

```
polynomial_resilience_set(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sig1_values,
sig2_values, alpha = 0.10)
```

Arguments

s0.A	Vector of surrogate values in the control group of Study A.
y0.A	Vector of primary outcome values in the control group of Study A.
s1.A	Vector of surrogate values in the treatment group of Study A.
y1.A	Vector of primary outcome values in the treatment group of Study A.
s0.B	Vector of surrogate values in the control group of Study B.
s1.B	Vector of surrogate values in the treatment group of Study B.
sig1_values	Vector of values of σ^2_{11} for which the probability of surrogate paradox should be calculated with the given data.
sig2_values	Vector of values of σ^2_{22} for which the probability of surrogate paradox should be calculated with the given data.
alpha	Threshold value of the surrogate paradox. Default is 0.10.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

Returns a ggplot object with values of σ_{11}^2 on the x-axis and σ_{22}^2 on the y-axis, and regions where $P(\Delta_B) < \alpha$ with given data and parameters highlighted in blue.

Author(s)

Emily Hsiao

References

Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Examples

```
n_A=200
n_B=200
s0.A <- rnorm(n_A, mean = 10, sd = 2)
 y0.A <- 5 + 0.5 * s0.A + rnorm(n_A, mean = 0, sd = 0.5)

s1.A <- rnorm(n_A, mean = 12, sd = 2.5)
 y1.A <- 5.5 + 0.6 * s1.A + rnorm(n_A, mean = 0, sd = 0.6)

s0.B <- rnorm(n_B, 10, 2)
s1.B <- rnorm(n_B, 11, 3)


sigma1_values <- seq(0.01, 1.2, length.out = 30)
sigma2_values <- seq(0.01, .4, length.out = 30)

  polynomial_resilience_set(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sigma1_values,
  sigma2_values, alpha = 0.05)
```

Q *Helper function for monotonicity test*

Description

Helper function for monotonicity test; should not be called directly by user.

Usage

```
Q(r, s, X)
```

Arguments

r	Index to start summation.
s	Index to end summation.
X	Vector of X values over which to sum.

Value

Q

Author(s)

Emily Hsiao

References

Hall, Peter, and Nancy E. Heckman. "Testing for monotonicity of a regression mean by calibrating for linear functions." Annals of Statistics (2000): 20-39.

S *Helper function*

Description

Helper function for monotonicity test.

Usage

```
S(a, b, r, s, X, Y)
```

Arguments

`a`	Value of a (regression coefficient)
`b`	Value of b (regression coefficient)
`r`	Index to start summation
`s`	Index to end summation
`X`	Vector of X values
`Y`	Vector of Y values

Value

Mean-squared error

Author(s)

Emily Hsiao

sd_test *Stochastic dominance test function*

Description

Runs the test of stochastic dominance given two vectors of surrogate values.

Usage

```
sd_test(s0, s1, alpha = 0.05)
```

Arguments

`s0`	Vector of surrogate values in control group.
`s1`	Vector of surrogate values in treatment group.
`alpha`	Desired alpha level of hypothesis test.

Value

`s_hat`	Value of test statistic
`p.value`	p-value of test
`reject`	Rejection decision of test

Author(s)

Emily Hsiao

References

Barrett, Garry F., and Stephen G. Donald. "Consistent tests for stochastic dominance." Econometrica 71.1 (2003): 71-104.

smoother_fitter *Helper function*

Description

Nadaraya-Watson kernel smoother

Usage

```
smoother_fitter(X, Y, kernel = gaussian_kernel, h)
```

Arguments

`X`	Vector of X values
`Y`	Vector of Y values
`kernel`	Kernel to use in the kernel smoother; defaults to Gaussian kernel
`h`	Bandwidth

Value

Returns a function which is the smoothed function; input takes in an x value.

Author(s)

Emily Hsiao

`test_assumptions` *Test assumptions to prevent surrogate paradox*

Description

Tests the assumptions necessary to prevent the surrogate paradox: stochastic dominance of surrogate values in the treatment group over control group, monotonicity of the relationship between surrogate and primary endpoint in both treatment and control group, and non-negative residual treatment effect of the treatment group over the control group. For computational efficiency, Version 2.0 of this package uses the monotonicity_test function from the MonotonicityTest package.

Usage

```
test_assumptions(s0 = NULL, y0 = NULL, s1 = NULL, y1 = NULL, trim = 0.95,
alpha = 0.05, type = "all", all_results = TRUE, direction = "positive",
monotonicity_bootstrap_n = 100, nnr_bootstrap_n = 200)
```

Arguments

`s0`	Vector of surrogate values in control group.
`y0`	Vector of primary endpoint values in control group.
`s1`	Vector of surrogate values in treatment group.
`y1`	Vector of primary endpoint values in treatment group.
`trim`	Proportion of data to keep after trimming the outliers. Defaults to 95%. Trims data by sorting by surrogate value and removing (1 - trim)/2 % of the lowest and highest surrogate values with their corresponding primary endpoint values.
`alpha`	Desired alpha level of tests.
`type`	Type of test to run. Defaults to "all"; possible inputs are "sd" (stochastic dominance), "monotonicity" (monotonicity), and "nnr" (non-negative residual treatment effect).
`all_results`	TRUE or FALSE; return all outputs from hypothesis tests. Defaults to TRUE.
`direction`	Direction of the test. Defaults to "positive", which tests that the treatment group stochastically dominates the control group, that μ_0 and μ_1 are monotonically increasing, and that $\mu_0 \leq \mu_1 \forall s$. Parameter "negative" tests that the control group stochastically dominates the treatment group, that μ_0 and μ_1 are monotonically decreasing, and that $\mu_1 \leq \mu_0 \forall s$.
`monotonicity_bootstrap_n`	Number of bootstrap samples for monotonicity test.
`nnr_bootstrap_n`	Number of bootstrap samples for nnr test.

Value

`result` Table or string of results of the tests

`sd_result` Detailed results of stochastic dominance test; only returned if all_results is TRUE

`monotonicity0_result` Detailed results of monotonicity test in control group; only returned if all_results is TRUE

`monotonicity1_result` Detailed results of monotonicity test in treatment group; only returned if all_results is TRUE

`nnr_result` Detailed results of nnr test; only returned if all_results is TRUE

Author(s)

Emily Hsiao

References

Barrett, Garry F., and Stephen G. Donald. "Consistent tests for stochastic dominance." Econometrica 71.1 (2003): 71-104.

Hall, Peter, and Nancy E. Heckman. "Testing for monotonicity of a regression mean by calibrating for linear functions." Annals of Statistics (2000): 20-39.

Hsiao, Tian, Parast. "Avoiding the Surrogate Paradox: An Empirical Framework for Assessing Assumptions." 2025 (Under Review)

Examples

```
m_c <- function(s) 1 + 2 * s
m_t <- function(s) 1 + 2 * s

s_c <- rnorm(100, 3, 1)
y_c <- sapply(s_c, function(s) rnorm(1, m_c(s), 1))
s_t <- rnorm(100, 3, 1)
y_t <- sapply(s_t, function(s) rnorm(1, m_t(s), 1))

test_assumptions(
s0 = s_c, y0 = y_t, s1 = s_t, y1 = y_t, type = "sd"
)


test_assumptions(
s0 = s_c, y0 = y_t, s1 = s_t, y1 = y_t, type = "all")
```

`T_m` *Helper function for monotonicity test*

Description

Calculates the test statistic

Usage

```
T_m(m, X, Y)
```

Arguments

`m`	m window size
`X`	Vector of X values
`Y`	Vector of Y values

Value

`stat`	Value of the test statistic
`stat_vals`	Vector of statistics before taking maximum
`b_vals`	Values of b
`Q_vals`	Values of Q

Author(s)

Emily Hsiao

Index